

REPORT DOCUMENTATION PAGE				Form Approved OMB No. 0704-0188	
The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing the burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.					
PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.					
1. REPORT DATE (DD-MM-YYYY) 26-11-2012		2. REPORT TYPE Final Technical		3. DATES COVERED (From - To) From 10/31/2009 to 10/31/2012	
4. TITLE AND SUBTITLE Numerical solution of Hamilton-Jacobi equations in high dimension				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER FA9550-10-1-0029	
				5c. PROGRAM ELEMENT NUMBER	
				5d. PROJECT NUMBER	
6. AUTHOR(S) Maurizio Falcone				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Dipartimento di Matematica SAPIENZA-Universita di Roma P. Aldo Moro, 2 00185 ROMA				8. PERFORMING ORGANIZATION REPORT NUMBER AH930	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) AFOSR 875 N. Randolph St Arlington, VA 22203				10. SPONSOR/MONITOR'S ACRONYM(S)	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S) AFRL-OSR-VA-TR-2012-1241	
12. DISTRIBUTION/AVAILABILITY STATEMENT A = Approved for public release; distribution is unlimited					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT The solution of nonlinear optimal control problems and the computation of optimal control laws is a difficult task that requires a fast response in many real applications. During the three years of the AFOSR grant we attacked the problem via the numerical solution of Hamilton-Jacobi equations describing the value function. In the framework of viscosity solutions, the value function is uniquely characterized and it allows for the computation of optimal controls in feedback form. Unfortunately, from a computational point of view, solving real problems in high dimension via Hamilton-Jacobi equations is still a huge task and efficient algorithms are required. Our research has been devoted to the development of new Fast Marching methods and domain decomposition techniques (or a combination of them) in order to improve CPU times, avoid useless computations and reduce memory requirements.					
15. SUBJECT TERMS Numerical methods, Optimal control, Pursuit-Evasion Games, Hamilton-Jacobi Equations, Fast Marching Methods, Domain Decomposition, Parallel Algorithms					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT unclassified(U)	18. NUMBER OF PAGES 21	19a. NAME OF RESPONSIBLE PERSON Maurizio Falcone
a. REPORT unclassified	b. ABSTRACT unclassified	c. THIS PAGE unclassified			19b. TELEPHONE NUMBER (Include area code) +39(6) 499123279

M. Falcone

INSTRUCTIONS FOR COMPLETING SF 298

1. REPORT DATE. Full publication date, including day, month, if available. Must cite at least the year and be Year 2000 compliant, e.g. 30-06-1998; xx-06-1998; xx-xx-1998.

2. REPORT TYPE. State the type of report, such as final, technical, interim, memorandum, master's thesis, progress, quarterly, research, special, group study, etc.

3. DATES COVERED. Indicate the time during which the work was performed and the report was written, e.g., Jun 1997 - Jun 1998; 1-10 Jun 1996; May - Nov 1998; Nov 1998.

4. TITLE. Enter title and subtitle with volume number and part number, if applicable. On classified documents, enter the title classification in parentheses.

5a. CONTRACT NUMBER. Enter all contract numbers as they appear in the report, e.g. F33615-86-C-5169.

5b. GRANT NUMBER. Enter all grant numbers as they appear in the report, e.g. AFOSR-82-1234.

5c. PROGRAM ELEMENT NUMBER. Enter all program element numbers as they appear in the report, e.g. 61101A.

5d. PROJECT NUMBER. Enter all project numbers as they appear in the report, e.g. 1F665702D1257; ILIR.

5e. TASK NUMBER. Enter all task numbers as they appear in the report, e.g. 05; RF0330201; T4112.

5f. WORK UNIT NUMBER. Enter all work unit numbers as they appear in the report, e.g. 001; AFAPL30480105.

6. AUTHOR(S). Enter name(s) of person(s) responsible for writing the report, performing the research, or credited with the content of the report. The form of entry is the last name, first name, middle initial, and additional qualifiers separated by commas, e.g. Smith, Richard, J, Jr.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES). Self-explanatory.

8. PERFORMING ORGANIZATION REPORT NUMBER. Enter all unique alphanumeric report numbers assigned by the performing organization, e.g. BRL-1234; AFWL-TR-85-4017-Vol-21-PT-2.

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES). Enter the name and address of the organization(s) financially responsible for and monitoring the work.

10. SPONSOR/MONITOR'S ACRONYM(S). Enter, if available, e.g. BRL, ARDEC, NADC.

11. SPONSOR/MONITOR'S REPORT NUMBER(S). Enter report number as assigned by the sponsoring/monitoring agency, if available, e.g. BRL-TR-829; -215.

12. DISTRIBUTION/AVAILABILITY STATEMENT. Use agency-mandated availability statements to indicate the public availability or distribution limitations of the report. If additional limitations/ restrictions or special markings are indicated, follow agency authorization procedures, e.g. RD/FRD, PROPIN, ITAR, etc. Include copyright information.

13. SUPPLEMENTARY NOTES. Enter information not included elsewhere such as: prepared in cooperation with; translation of; report supersedes; old edition number, etc.

14. ABSTRACT. A brief (approximately 200 words) factual summary of the most significant information.

15. SUBJECT TERMS. Key words or phrases identifying major concepts in the report.

16. SECURITY CLASSIFICATION. Enter security classification in accordance with security classification regulations, e.g. U, C, S, etc. If this form contains classified information, stamp classification level on the top and bottom of this page.

17. LIMITATION OF ABSTRACT. This block must be completed to assign a distribution limitation to the abstract. Enter UU (Unclassified Unlimited) or SAR (Same as Report). An entry in this block is necessary if the abstract is to be limited.

Final report of the Project
**Numerical solution of Hamilton-Jacobi
equations in high dimension**
AFOSR contract n. FA9550-10-1-0029

Maurizio Falcone
Dipartimento di Matematica
SAPIENZA-Università di Roma
P. Aldo Moro, 2 - 00185 ROMA
email: `falcone@mat.uniroma1.it`

November 23, 2012

Abstract

The solution of nonlinear optimal control problems and the computation of optimal control laws is a difficult task that requires a fast response in many real applications. During the three years of the AFOSR grant we attacked the problem via the numerical solution of Hamilton-Jacobi equations describing the value function. In the framework of viscosity solutions, the value function is uniquely characterized and it allows for the computation of optimal controls in feedback form. Unfortunately, from a computational point of view, solving real problems in high dimension via Hamilton-Jacobi equations is still a huge task and efficient algorithms are required. Our research has been devoted to the development of new Fast Marching methods and domain decomposition techniques (or a combination of them) in order to improve CPU times, avoid useless computations and reduce memory requirements.

1 Introduction

The computation of optimal controls is one of the primary goals in control theory. The classical approach based on Pontryagin's Maximum Principle leads to a two point boundary value problem for a system of ordinary differential equations describing the dynamics and the co-state and provide necessary conditions for optimality. Although some numerical methods have been proposed for

that problem, it is well known that this approach is limited by the following difficulties: a) the initialization of the co-state equations to start the numerical procedure can be very difficult and often requires weeks to be solved, b) the optimal control which is obtained is open-loop and c) the optimal state-costate couple just satisfies necessary conditions so there is no guarantee that it corresponds to a global minimum for the cost functional on the space of admissible controls. Despite those limitations, the approach based on the Pontryagin's Maximum Principle has been up to now the only approach which has produced feasible solutions for real industrial problems.

The second classical approach to the solution of optimal control problems is Dynamic Programming, introduced by R. Bellman in the 60's [7]. In this approach, a central role is played by the value function u of the problem, defined as

$$u(x) = \inf_{a(\cdot) \in A} J(x, a(\cdot))$$

and the reconstruction of control is done starting from the Bellman equation associated to the problem once the value function has been found (here x will represent the initial position of the system, A will be the set of admissible controls and J is the cost functional).

The development of the theory of viscosity solutions in the last twenty years has finally given a general framework for the characterization of the value function in all the classical problems of deterministic and stochastic control theory (finite horizon, infinite horizon, minimum time, optimal switching, impulsive control, pursuit-evasion games). Indeed, under very general assumptions, one is able to prove that the unique viscosity solution of the Bellman (resp. Isaacs) equation associated to the control (resp. game) problem is the value function. The advantage of this approach is twofold. It gives a precise characterization of the global optimum for the problem and it allows to obtain optimal controls in feedback form. This has motivated a large research activity on the development of numerical methods to solve the Bellman (Isaacs) equation as well as on the numerical synthesis of feedback controls based on the knowledge of an approximate value function.

The main drawback of this approach is due to the well known "curse of dimensionality" of Dynamic Programming. Indeed, the numerical solution of a nonlinear partial differential equation in high-dimension is a difficult task and requires new ideas and algorithms. This project had the aim of developing new methods coupling the semi-Lagrangian (or Dynamic Programming) approximation with Fast Marching and domain decomposition techniques in order to solve optimal control and pursuit-evasion games.

2 Background

In this section we briefly review, for the reader's convenience, some classical topics underlying the new methods we developed during the three years of the AFOSR grant.

2.1 Semi-Lagrangian schemes for control problems and games via Dynamic Programming equations

The Dynamic Programming approach is based on the characterization of the value function of the problem via the Bellman equation, a nonlinear first order partial differential equation for deterministic optimal control and game problems. For instance, the Hamilton-Jacobi-Bellman (HJB) equation for minimum time problems (with target set \mathcal{T}) is

$$\begin{cases} \max_{a \in A} \{-f(x, a) \cdot \nabla u(x)\} - 1 = 0 & x \in \mathbb{R}^n \setminus \mathcal{T}, \\ u(x) = g(x) & x \in \partial\mathcal{T}. \end{cases} \quad (1)$$

This leads to the characterization of the optimal control in feedback form, which requires the exact (or approximate) knowledge of u :

$$a^*(x) = \arg \min_{a \in A} \{f(x, a) \cdot \nabla u(x)\}, \quad x \in \mathbb{R}^n \setminus \mathcal{T}. \quad (2)$$

Similarly, the Hamilton-Jacobi-Isaacs (HJI) equation for pursuit-evasion games is

$$\begin{cases} \min_{b \in B} \max_{a \in A} \{-f(x, a, b) \cdot \nabla u(x)\} - 1 = 0 & x \in \mathbb{R}^n \setminus \mathcal{T} \\ u(x) = g(x) & x \in \partial\mathcal{T} \end{cases} \quad (3)$$

and the corresponding optimal feedback strategies can be computed from u by

$$(a^*(x), b^*(x)) = \arg \max_{b \in B} \min_{a \in A} \{f(x, a, b) \cdot \nabla u(x)\}, \quad x \in \mathbb{R}^n \setminus \mathcal{T}. \quad (4)$$

Let us introduce the basic concepts of Semi-Lagrangian (SL) schemes that we will use as a building block for our methods.

Consider a structured grid G and denote its nodes by x_i , $i = 1, \dots, N$. The space step is assumed to be uniform and equal to $\Delta x > 0$.

The HJB equation (1) can be discretized by means of a discrete Dynamic Programming Principle, so that the relationship with the optimal control framework is never lost. Standard arguments [5] lead to the following scheme (similar arguments apply for the HJI equation (3)):

$$u(x_i) = \min_{a \in A} \left\{ u(x_{i,a}) + \frac{|x_i - x_{i,a}|}{|f(x_i, a)|} \right\}, \quad (5)$$

where $x_{i,a}$ is a *non-grid* point, obtained by integrating, until a certain final time τ , the ordinary differential equation

$$\begin{cases} \dot{y}(t) = f(y, a), & t \in [0, \tau] \\ y(0) = x_i \end{cases} \quad (6)$$

and then setting $x_{i,a} = y(\tau)$. To make the scheme fully discrete, the set of admissible controls A is discretized in N_c points and the minimum in (5) is computed by a direct comparison of the N_c values.

Depending on the choice of τ , on how equation (6) is solved and on how $u(x_{i,a})$ is interpolated, we get different versions of the semi-Lagrangian scheme (5).

Here we list two different discretizations that often appear in the literature:

- 2-points SL. This scheme is used, for example, in [26] and [27]. Equation (6) is solved by an explicit forward Euler scheme until the solution intercepts the line connecting two neighbouring points $x_{i,1}$ and $x_{i,2}$ (see Figure 1a). The value $u(x_i)$ is computed by linear interpolation of the values $u(x_{i,1})$ and $u(x_{i,2})$.
- 3-points SL. This scheme is used, for example, in [11]. Equation (6) is solved by an explicit forward Euler scheme until the solution is at distance Δx from x_i , where it falls inside the triangle of vertices $x_{i,1}$, $x_{i,2}$, and $x_{i,3}$ (see Figure 1b). The value $u(x_i)$ is computed by linear interpolation of the values $u(x_{i,1})$, $u(x_{i,2})$ and $u(x_{i,3})$.

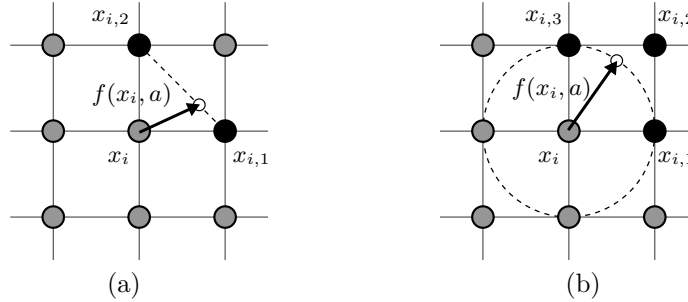


Figure 1: (a) 2-points SL scheme, (b) 3-points SL scheme.

2.2 Fast Marching Method for the Eikonal equation

The Fast Marching Method (FMM) has been introduced by Sethian [25] in 1996 as a fast solver for the eikonal equation:

$$\begin{cases} c(x)|\nabla u(x)| = 1 & x \in \mathbb{R}^n \setminus \mathcal{T}, \\ u(x) = g(x) & x \in \partial\mathcal{T}, \end{cases}$$

where c, g are Lipschitz functions and the target set \mathcal{T} is a closed subset.

Unlike more classical fixed-point methods that simply iterate “brute force” on the whole grid until convergence, the main idea of FMM is to visit grid nodes in a dynamic-dependent order exploiting the hyperbolic nature of the equation, namely trying to follow the characteristic flow emanating from the target set \mathcal{T} . To this end grid nodes are iteratively divided in three subsets: *Accepted* (*ACC*) region, *Narrow Band* (*NB*) region and *Far* (*FAR*) region (see Figure 2). Nodes in *ACC* are definitively computed, nodes in *NB* are computed but their values are not yet final and nodes in *FAR* are not yet computed.

At the beginning *ACC* only consists of nodes belonging to the target set \mathcal{T} , whose values are imposed using the Dirichlet condition given by g . Then the algorithm finds itself a correct order for processing all the remaining nodes. That

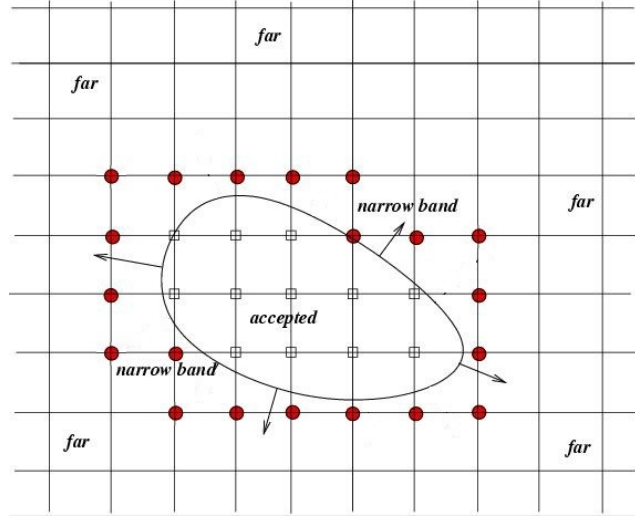


Figure 2: Partition of grid nodes in *Accepted*, *Narrow Band* and *Far*.

order is found in such a way a *causality* principle is respected, i.e. the value of a node is labelled as “final” only if it depends on already “final” values and it cannot be affected by the future computation.

In practice, the node in *NB* with the *minimal value* enters *ACC* and its first neighbours are computed by the underlying scheme entering *NB* (if not yet in). We remark that this *minimal value rule* corresponds to compute the solution of the equation following its level sets. Indeed, suppose for simplicity that $g = 0$, so that $\partial\mathcal{T}$ is the zero level set of u . Then, the node in *NB* with minimal value identifies the minimal variation of the solution, hence the direction of its gradient. It follows that *NB* expands under the gradient flow of the solution itself, which is exactly equivalent to say that *NB* is, at each step, an approximation of a level set. In the case of the Eikonal equation, the gradient of the solution coincides with the characteristic field of the equation, hence FMM computes the correct solution, namely the same solution obtained with a full grid method employing the same scheme.

More in general, FMM still works for problems with mild anisotropy, where gradient lines and characteristics define small angles and lie, at each point, in the same simplex of the underlying grid.

On the other hand, when a strong anisotropy comes into play, as for the anisotropic Eikonal equation studied in [26], FMM fails and there is no way to compute the solution following its level sets.

FMM is an example of what is called a *local single-pass* scheme, where *local* refers to the fact that the computation of a grid point involves only the values of neighbouring points, the region *NB* is only one-cell or two-cell large and no information coming from *FAR* region is used. On the other hand *single-pass* means that each grid point is re-computed at most r times, where r only depends on the considered equation and the grid structure, not on the number of

grid points.

2.3 Domain Decomposition for Hamilton-Jacobi equations

The Domain Decomposition is a classical and quite general technique to solve a problem on a huge grid by dividing it into sub-problems of a reasonable size. The idea is rather simple and intuitive: the domain of computation is split in several sub-domains and each sub-domain is assigned to a processor, so that the whole computation is carried on in parallel (see for instance [24]). For an application of this technique in the context of Hamilton-Jacobi equations we refer the reader to [12, 18]. The delicate point for the efficiency of this kind of parallel methods is the fact that processors must share some parts of their sub-domains (at least the sub-domain boundaries) and must communicate during the computation using transmission conditions (see Figure 3).

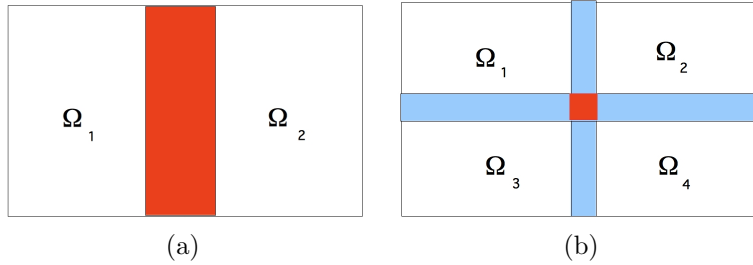


Figure 3: Classical domain decomposition with overlapping boundaries: (a) 2 sub-domains, (b) 4 sub-domains.

Since, in general, there is no a-priori knowledge on where the correct information will flow from, due to the arbitrariness of the decomposition, these communications must take place at every iteration. This is a real bottleneck for the parallel computation, since each processor continues working until a global convergence condition is satisfied.

3 Objectives

Let us recall the original objectives of this research project:

3.1 First year

Analysis and development of Fast Marching methods for first order Hamilton-Jacobi equation with convex Hamiltonians. Applications to optimal control problems in low dimension.

3.2 Second year

Extension of Fast Marching methods to high-dimensional control problems. Analysis of convergence, a-priori error estimates. Implementation of the algorithms and coupling with the domain decomposition technique. Parallel algorithms.

3.3 Third year

Extension of Fast Marching methods to non-convex Hamiltonians and applications to differential games. Analysis of convergence, a-priori error estimates. Implementation of the algorithms and coupling with the domain decomposition technique. Parallel algorithms.

4 Findings

The results that have been obtained during this project are mainly related to the development of new ideas and algorithms. Only in few cases we obtained a complete analysis of the new methods, giving new bounds for their complexity and acceleration. However, it is interesting to note that the general bounds for global semi-lagrangian methods (see e.g. [13, 14]) still apply. The extensive testing of our methods on a large number of benchmarks shows that the methods are really effective and improve the results in the literature.

4.1 A Parallel Fast Marching Method for the Eikonal equation

The classical FMM described in Section 2.2 works in a highly sequential way, due to the fact that the *minimal value rule* accepts only one grid node in NB at a time. This is the main reason explaining the scarce literature on parallelization techniques for FMM. To our best knowledge, only few contributions are available on this topic (see for instance [20, 28, 29]) dealing with parallel algorithms based on domain decomposition methods, in some cases optimized for Graphics Processing Units.

In this direction we developed a parallelization approach for FMM which is not based on a domain decomposition technique and it is also easier to implement than Hermann's method proposed in [20].

The main idea is to split among threads only the part of the boundary of the domain where the Dirichlet conditions are given, rather than splitting the whole computational domain. The sub-domains resulting from this splitting are then used by each thread as starting points for computing independent solutions. Thereby, the interactions among threads are realized by a relatively simple procedure which guarantees the correct order of acceptance and keeps the overlapping areas to the minimum. In the end, the computational domain is automatically divided among threads in a way which depends on the dynamics of the problem. Several tests showed that our method is computationally competitive

with Hermann’s method, in some cases even more efficient.

Figure 4 shows a comparison of the speed up for our algorithm and Hermann’s method applied to the Eikonal equation.

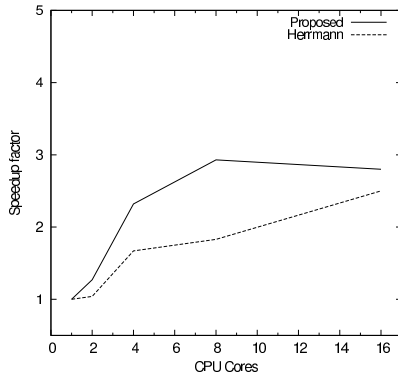


Figure 4: Speedup comparison.

In this test the domain is a square and boundary conditions are imposed on a small circle located in a corner of the domain. The pre-determined domain decomposition employed by Hermann splits the domain into equal parts. This choice is clearly not optimal, since at the beginning of the computation only the thread assigned to the sub-domain containing the circle is really active. Our algorithm is able to assign a job to each core from the very beginning and never degenerates into a serial mode. We refer the reader to [P1] for further details.

4.2 Progressive Fast Marching Method (PFMM) for control problems and games

Several efforts have been made to extend the classical FMM, in order to increase its efficiency and manage more general equations/problems (see e.g. [22] and the references therein). As described in Section 2.2, FMM is designed on the Eikonal equation, exploiting the fact that the characteristic field of the equation coincides with the gradient of the solution itself. This property does not hold for general Hamilton-Jacobi equations, hence FMM fails in computing the correct solution, especially in problems with strong anisotropy. To our opinion, the most valuable work aiming at overcoming this difficulties is contained in the paper [26], where an Ordered Upwind Method has been proposed to solve anisotropic equations in a Fast Marching fashion. This method produces good results for several difficult tests, but the spirit of FMM is somehow betrayed, in the sense that the locality property of the method is completely lost. Moreover, the computed solution does not coincide with the solution obtained by a full grid method using the same scheme. The error depends on the anisotropy of the problem and vanishes only in the limit, as the space step of the grid goes to zero.

In this direction we investigated the possibility of keeping the locality and single-pass properties of FMM, but replacing the minimal value rule as acceptance criterium for nodes in NB .

In the framework of optimal control, we analyzed the behavior of the approximate value function solving the HJB equation (1) subject to local perturbations, aiming at understanding how NB nodes depend on the already accepted ones. We observed that, over a certain threshold of the perturbation, only a small number of grid points of the narrow band do not vary, and this suggests that they have to be processed before the others.

To be more precise, consider the situation sketched in Figure 5a. Red nodes are in ACC , green nodes are in NB and black nodes are in FAR .

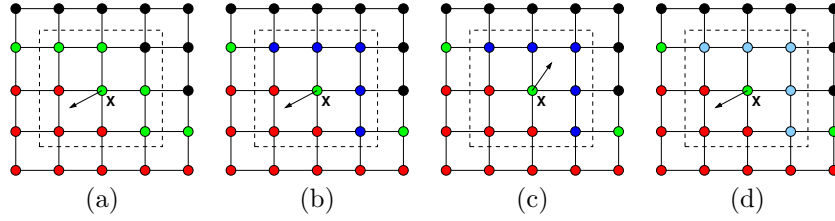


Figure 5: Dependency of NB nodes on local perturbations of the value function.

For each $x \in NB$ consider its first neighbors, namely the nodes needed to compute the value $u(x)$. The arrow represents the optimal dynamics at x in the HJB equation (1). Now replace the values of *all* the *not-accepted* neighbors of x by a tentative value \bar{u} (blue nodes in Figure 5b). A *small* tentative value can change the optimal control and then the value $u(x)$ (see Figure 5c). But, if we *progressively* increase this tentative value, it can happen that, over a certain threshold (see light blue nodes in Figure 5d), the “new” optimal control is no longer favorable and the scheme comes back to choose the “original” optimal control which only depends on accepted nodes. In this way, we actually mimic the influence that still unknown values (NB and FAR neighbors) can have on $u(x)$ in the future. Then, we define the *threshold* $\sigma(x)$ as the minimal tentative value such that $u(x)$ *does not change* for any $\bar{u} \geq \sigma(x)$. This threshold quantifies the dependency of $u(x)$ on its not-accepted neighbors and then it appears natural to consider the node in NB with minimal threshold as the most correct. This argument produces a new criterium for ordering the grid nodes (different from the minimal value rule), showing the ability to compute the correct solution for a wide class of Hamilton-Jacobi equations, also in the case of nonconvex Hamiltonians, e.g. for Hamilton-Jacobi-Isaacs equations related to Pursuit-Evasion games. Moreover, the Progressive Fast Marching Method (PFMM) is a coherent extension of the Fast Marching method, since it gives back the classical FMM when applied to isotropic equations of eikonal type. Indeed, in these cases the node with minimal threshold is also the node with minimal value. We refer the reader to [P2] for further details.

Figure 6 presents the result of PFFM compared with the classical FMM in solv-

ing the anisotropic Eikonal equation, that can be re-casted as an HJB equation by choosing in (1)

$$f(x, y, a) = a(1 + (\lambda a_1 + \mu a_2)^2)^{-\frac{1}{2}}$$

$$\mu, \lambda > 0, \quad a = (a_1, a_2) \in A = B(0, 1), \quad \mathcal{T} = \{(0, 0)\}.$$

In Figure 6a-b we show the level sets of the wrong solution computed by FMM against the correct solution computed by PFFM, in Figure 6c-d we show the corresponding orders of acceptance at an intermediate stage.

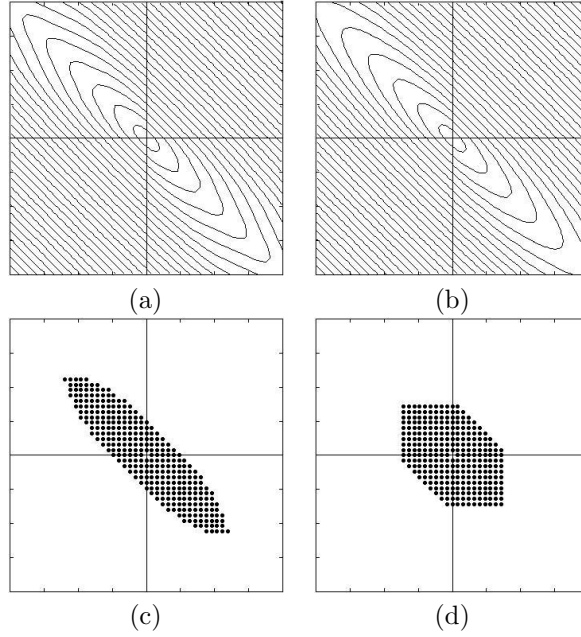


Figure 6: Anisotropic Eikonal equation. Level set of the solution: (a) FMM, (b) PFFM. Order of acceptance: (c) FMM, (d) PFFM.

Figure 7 presents the result of PFFM applied to a pursuit-evasion game in the plane with control constraints. Using reduced coordinates (see [17] for details) we can re-cast the problem as an HJI equation by choosing in (3)

$$f(x, y, a, b) = v_P a - v_E b, \quad v_P = 2, v_E = 1, \quad \mathcal{T} = \{|x - y| < \varepsilon\}, \quad \varepsilon > 0,$$

$$a \in A = B(0, 1), \quad b \in B = \{(\cos \theta, \sin \theta) : \theta \in [\pi/4, 7\pi/4]\}.$$

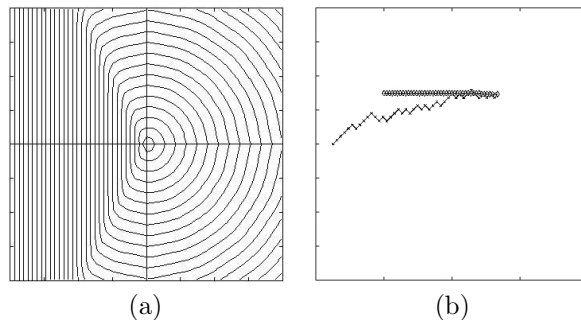


Figure 7: Pursuit-Evasion game with control constraints. (a) level set of the solution, (b) a couple of optimal trajectories.

4.3 Patchy Domain Decomposition (PDD) for a class of Hamilton-Jacobi-Bellman equations

In the spirit of the Domain Decomposition technique described in Section 2.3, we developed a new method for the parallelization of algorithms solving the HJB equation (1). The new method is inspired by some works of Ancona and Bressan (see [1, 2, 3]) related to the problem of the asymptotic stabilization of a control system. In these works the authors introduced the “patchy” technique, based on the division of the domain in regions (patches) which are invariant with respect to the optimal feedback dynamics of the problem. Keeping this idea in mind, we ended up with a dynamical construction of a domain decomposition that exploits the invariance of the sub-domains, assigning each patch to a processor. The resulting parallel algorithm is very efficient, since it avoids the communication costs between processors.

The main steps of the construction are the following: we first solve the equation on a coarse grid by means of the classical (static) domain decomposition technique. Then we compute an approximation of the optimal feedback control on the actual (fine) grid using in (2) a simple linear interpolation of the coarse solution. Finally, we define a partition of the target set and we let flow each part according to the approximate optimal dynamics. In practice each patch is an approximation of a tube of characteristic lines emanating from the target set. We refer the reader to [P3] for a detailed description of the patchy domain decomposition.

In Figure 8 we show the invariant patchy decompositions obtained by our algorithm applied to the following classical problems:

- (a) Eikonal dynamics $f(x, y, a) = a$, $A = B(0, 1)$ and $\mathcal{T} = B(0, 0.5)$.
- (b) Fan dynamics $f(x, y, a) = (|x + y| + 0.1)a$, $A = B(0, 1)$ and $\mathcal{T} = \{x = 0\}$.
- (c) Zermelo dynamics $f(x, y, a) = 2.1a + (2, 0)$, $A = B(0, 1)$ and $\mathcal{T} = B(0, 0.5)$.

- (d) Brockett dynamics $f(x, y, z, a) = (a_1, a_2, xa_2 - ya_1)$, $A = [-5, 5]^2$ and $\mathcal{T} = B(0, 0.25)$.

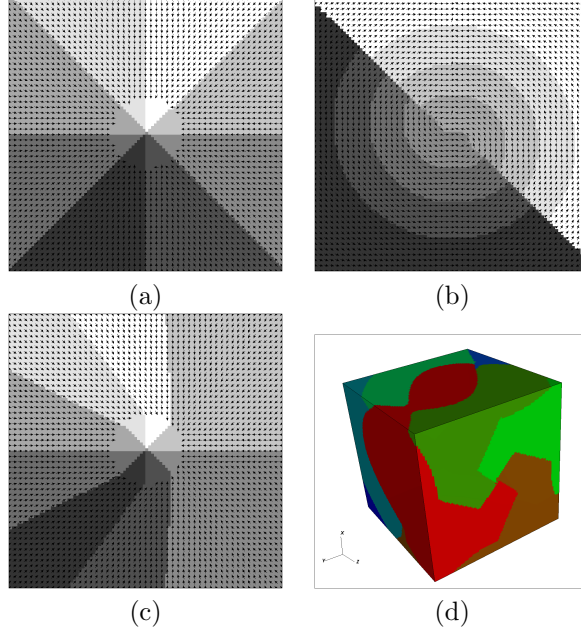


Figure 8: Patchy decompositions (with 8 patches) for some classical problems: (a) Eikonal, (b) Fan, (c) Zermelo, (d) Brockett. Optimal controls are overlaid for the first three problems.

We remark that the initial partition of the target could not be performed efficiently (or at all) for small sets, such as lines or points. In these cases we can always start our construction with a single patch and run the algorithm in serial mode until the patch reaches a reasonable size. Then the solution is computed in the patch and it is used as a new target to be divided, in order to restart the algorithm in parallel mode.

Similarly, in some problems the dynamics could induce very different sizes in the patches, resulting in a loss of performance for the parallel algorithm if one processor finishes his job before the others and there are no more patches in the cue to be assigned. This situation can be avoided by mixing iteratively the construction of the patches with the parallel computation of the solution: we first run the construction until the sizes of patches remain balanced, then we compute in parallel the solution in the patches and finally we restart the procedure considering the computed solution as a new target.

We also remark that our method could be applied in problems without target. Indeed, using the coarse solution, we can approximately localize the sources of characteristics, where we can put fictitious targets and then start the patchy decomposition.

The performance of the patchy domain decomposition is very satisfactory. Despite the additional computation needed to build the patches, the overall CPU time for computing the solution is, for all the test we performed, better than that of the classical domain decomposition, even better as we increase the number of patches. This is due not only to the absence of transmission conditions between processors, but also to a better distribution of the computational resources. Indeed, each processor never remains idle, when it finishes his job it can be immediately re-assigned to the computation in another patch. Moreover, if the patches are balanced in size, all the processors will actually stop at the same time. Nevertheless, comparing our solution and the solution of the classical domain decomposition, an additional error in the approximation appears, since the underlying structured grid cannot accurately follow the characteristic lines defining the boundary of the patches. To be precise, the “real” patches exist at the continuous level and what we compute is just their projection on the grid. Figure 9 shows the “patchy error” of the solutions for the dynamics (a),(b),(c) defined above.

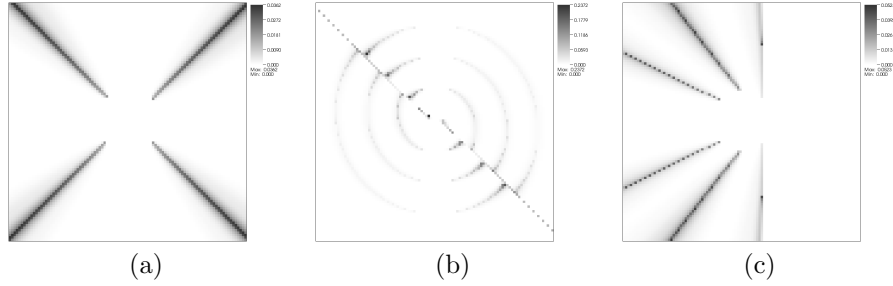


Figure 9: Patchy error for some classical problems: (a) Eikonal, (b) Fan, (c) Zermelo.

Surprisingly, the patchy error remains localized on the boundaries of the patches and, by numerical evidence, it decreases as the space step of the grid goes to zero. The theoretical proof of an error estimate of this kind is still under investigation.

4.4 Features and limitations of Local Single Pass schemes solving general Hamilton-Jacobi equations

Extending Fast Marching techniques to general Hamilton-Jacobi equations is still a challenging topic. In the context of control problems many numerical schemes were analyzed and employed (e.g. finite difference, semi-Lagrangian, discontinuous Galerkin and finite volumes) to produce fast methods that can be competitive with industrial solvers based on Pontryagin’s principle.

We recall that the ultimate goal is to overcome the well known “curse of dimensionality” related to the Bellman’s approach (Dynamic Programming Principle) for the solution of such problems. In high dimension, terribly huge computational efforts are the price one has to pay in order to easily get feedback controls

and optimal trajectories, against the open-loop controls and sub-optimal trajectories given by Pontryagin-like methods.

The common philosophy shared by Fast Marching methods is always to reduce memory allocation and CPU times, by means of *local single-pass* schemes which aim to

- keep the computational efforts strictly localized on regions which are considerably small with respect to the huge grids needed for the numerical discretization of the corresponding equations;
- avoiding useless computations minimizing the number of times each grid point is re-computed.

Beside the development of such fast solvers, including the methods described in the previous sections, we considered a deeper study [P4] of their features and limitations, aiming at answering the following question:

can *local single-pass* methods solve every Hamilton-Jacobi equation?

To this end we reviewed the most known (and efficient) methods falling in the class of Fast Marching solvers, trying to identify a comprehensive classification of the equations that they can actually solve.

Since the introduction of the classical Fast Marching Method (FMM) for the Eikonal equation, Hamilton-Jacobi equations have been divided in two classes: given a grid, we have

- (E) Eikonal-like equations, whose characteristic lines coincide or lie in the same simplex of the gradient lines of their solutions.
- (!E) Non eikonal-like equations, for which there exists at least a point where the characteristic line and the gradient of the solution diverge enough to not lie in the same simplex.

By construction (see Section 2.2) FMM clearly works for equations of type (E) and fails for equations of type (!E).

In the context of Hamilton-Jacobi-Bellman equations associated to minimum time problems with target, we ended up with another possible classification:

- (D) Equations with smooth characteristics. Information spreads from the target to the rest of the space along smooth lines, without shocks. The corresponding solution is differentiable.
- (!D) Equations with nonsmooth characteristics. Information starts from the target and then crashes, creating shocks. The corresponding solution is Lipschitz continuous.

We point out that the original FMM was proposed for an upwind finite difference discretization of the Eikonal equation. In particular, a safe implementation of the algorithm forces every node in NB to be computed by means of nodes in ACC only, possibly producing wrong values. Nevertheless, the node in NB with minimal value turns out to be the correct one, not affected by any other node in NB , hence truly dependent on nodes in ACC only.

On the other hand, the semi-Lagrangian schemes we always employ for solving Hamilton-Jacobi equations (which are more accurate due to their ability to follow directions not aligned to the grid) usually compute nodes in NB by using nodes in ACC and NB . In the end, the minimal value rule among nodes in NB still produces the correct solution and this implies that also in this case the correct node actually depends on nodes in ACC only.

This feature is crucial to understand a point of view that is (to our best knowledge) completely new in the community of Fast Marching methods: in general, the minimal value rule *is not* an essential request to make the algorithm work, rather it could be an obstacle in cases with strong anisotropy.

As remarked in Section 2.2, accepting the node in NB with minimal value corresponds to compute the solution following its level sets, a strategy that fails for equations belonging to the class (!E) (e.g. the anisotropic Eikonal equation). Thus FMM can only solve equations in the classes (E & D) or (E & !D).

On the contrary, equations falling in the class (!E & D) (but clearly also equations of type (E & D)) can be correctly solved, even faster than FMM, by simply accepting all the nodes in NB that depend on nodes in ACC only. Indeed, due to the assumption of being in the class (D) (i.e. there is no shock and then no crossing of information), these nodes have values that depend on information coming from one side only, driven by a smooth characteristic. They are computed starting from nodes in ACC belonging to the target \mathcal{T} , hence they are correct by induction.

Unfortunately, the most interesting (but also the most simple) equations associated to control problems are known to develop shocks, thus falling in the class (!D), among which equations of type (!E & !D) are even more interesting. In these cases the minimal value rule becomes essential to properly localize the shocks. Indeed, the fact that the narrow band NB expands as an approximation of a level set of the solution implies that a grid node close to a shock is reached from two different directions at the same time and its value is computed using correct information, possibly coming from both sides.

On the other hand, if NB is not an approximation of a level set (a necessary condition, as explained above, to deal with (!E) equations), then a grid node close to a shock could be reached by NB at two different times. In this situation it is impossible perceiving, using local information only, if the first arrival is the correct one. Whenever an error occurs, it is propagated and it cannot be redressed in the future, since the algorithm continues the enlargement of NB and ACC and never re-computes already accepted nodes.

Our conclusion, which is the main achievement of this study, is that in general equations of type (!E & !D) cannot be solved by local single-pass schemes. One has to either enlarge the width of the narrow band NB giving the local-

ity up, and/or re-compute the already accepted nodes giving the single-pass property up. In both cases the gain of Fast Marching methods with respect to memory allocation and efficiency is completely lost. In the worst situations, the narrow band NB can fill the whole domain and/or the number of times one node is re-computed can diverge, so that in general the convergence of the algorithm takes place only in the limit. In the end we recover the classical fixed-point full grid methods, that surely can solve every equation, even if they are terribly slow to be applied to real problems in high dimension.

To further convince the reader that equations of type (!E & !D) are the theoretical limit for the application of local single-pass methods, we finally remark that we found very pathological examples in our experiments. By coupling strong anisotropy and space inhomogeneity, we ended up with two kinds of (!E & !D) equations for which a local single pass scheme cannot even start, not only for the problem of localizing the shocks described above.

In the first case, there exists no node in NB that truly depends on nodes in ACC only: the discretization of the equation introduces, no matter how fine is the numerical grid, an artificial loop-dependency between nodes in NB that can be only solved by means of fixed-point iterations, thus breaking the single-pass property.

Even worst, in the second case, despite the narrow band is stabilized through fixed-point iterations, we still do not find exact values in NB (“exact” compared to solution of a full grid method based on the same numerical semi-Lagrangian scheme): the numerical grid introduces a kind of spurious reflux of information from the FAR region that has to be taken into account by enlarging the width of the narrow band, thus breaking the locality property of the method. Sometimes this reflux can be reduced by decreasing the space step of the grid, but in general it could disappear only in the limit.

4.5 Numerical approximation of Nash equilibria

The notion of Nash equilibrium is considered one of the most important achievements in the last century for its impact on the analysis of economic and social phenomena. While there is an extensive literature dealing with Nash equilibria for static non-cooperative games, the analysis of Nash equilibria for nonzero-sum multi-player differential games is more limited (see e.g. the monographs [6, 19] for an presentation of this theory). Moreover, only few papers give a characterization of the value functions for the players in terms of partial differential equations, as it was the case for control problems and for zero-sum differential games described for example in [5]. More precisely, we know that under suitable regularity assumptions, if the value functions of a non-cooperative nonzero-sum multi-player differential game exist, they satisfy a system of first-order Hamilton-Jacobi equations, see [4]. Typically theoretical results about these problems are valid only in very special cases and they are essentially limited to games in one dimension and with a simple dynamics, see e.g. [8, 9, 10]. More important, it is well known that the system of Hamilton-Jacobi equations can be ill-posed and that there are no theoretical results if the dimension of the

problem is greater than one.

From the numerical point of view the situation is even more deceiving since few results are available for Nash equilibria in the framework of non-cooperative games. In particular, we mention the paper [15] where an approximation of (static) Nash equilibria based on Newton methods is presented, and the paper [16] where the approximation is obtained via penalty methods.

In this context we proposed an approximation scheme for Nash equilibria of non-cooperative differential games. Following Bressan [8], we considered the system of stationary Hamilton-Jacobi equations giving a characterization of the value functions for a class of infinite-horizon games with nonlinear costs exponentially discounted in time. We extended to this system the class of dynamic programming approximation schemes studied for zero-sum differential games (see [17] for a detailed analysis of the schemes corresponding to pursuit-evasion games as well as some numerical tests). To our best knowledge, this is the first approximation scheme for computing Nash equilibria in the framework of differential games. Despite the lack of theoretical results, several tests showed that our scheme is convergent for some two-player games in one and two dimensions. In particular, in dimension one, we found that our numerical solution to the system of Hamilton-Jacobi equations gives an approximation of the “admissible solution” characterized by Bressan. We refer the reader to [P5] for details on the new method.

4.6 Coupling Value Iteration and Policy Iteration for solving Hamilton-Jacobi equations

The classical policy iteration method for control problems is known to have superlinear convergence in many interesting cases, provided the initial condition is sufficiently close to the solution (see the monograph [21] and the precise result in [23]). This limitation reduces the impact of the method and forces to use value iteration, which is more expensive in terms of CPU time but is globally convergent to the optimal solution under rather general assumptions.

We have developed a new method [P6] for the solution of Hamilton-Jacobi equations related to optimal control problems and games that tries to get the advantages of both method with an efficient coupling. The main idea is to start the computation using the value iteration method and then switch to the policy iteration method when a threshold fixed on the error is reached. The delicate point is to accurately determine this threshold, in order to avoid cumbersome computation due to the value iteration and, at the same time, to be reasonably sure that the policy iteration method will finally converge to the optimal solution. We analyzed several criteria for an efficient coupling in a number of examples in dimension two, three and four illustrating the properties of the new method.

5 Collaborators and contacts

Simone Cacace (post-doc paid by the AFOSR grant), Emiliano Cristiani (researcher), Athena Picarelli (PhD student), Dante Kalise (post-doc), Alessandro Alla (Phd student).

5.1 AFRL Points of Contact

Starting from the 2010 review meeting, held in in Washington DC (August 9-12, 2010), we had some contacts with Richard Erwin (richard.erwin@kirtland.af.mil) and with Prof. Meir Patcher (AFIT). R. Erwin visited our department on September 2011 to discuss some research topics relate to pursuit-evasion games.

6 Workshops and conferences

Beside the annual review meetings at Arlington, the results obtained during this project have been presented in several invited talks at international conferences:

1. 1st Workshop on Computational Issues in Nonlinear Control, Monterey, November 2009
2. EPSRC Symposium on Game Theory for Finance, Social and Biological Sciences, Warwick, April 2010
3. ICIAM 2011 - Minisymposium “Efficient Numerical Methods For Hamilton-Jacobi Equations”, Organizers: O. Bokanowski and M. Falcone, Vancouver, July 2011
4. SIAM Conference on Control and its Applications, Baltimore, July 2011
5. IFAC 2011, Milan, September 2011
6. SADCOWorkshop “Optimal Control”, London, September, 2011
7. Workshop “Control and Optimization of PDEs”, Graz, October, 2011
8. 2nd Workshop on Computational Issues in Nonlinear Control, Monterey, November 2011
9. SADCOWorkshop & Spring School “Applied and Numerical Optimal Control”, Paris, April 2012
10. Minisymposium at ORCOS 2012 “Optimal Control, Dynamic Games and Nonlinear Dynamics” Vienna University of Technology, June, 2012.

7 Publications

Here we list the publication produced during the three years of the AFOSR project.

- [P1] M. BREUSS, E. CRISTIANI, P. GWOSDEK AND O. VOGEL *An adaptive domain-decomposition technique for parallelization of the Fast Marching method*, Appl. Math. Comput., 218 (2011), pp. 32–44
- [P2] S. CACACE, E. CRISTIANI, AND M. FALCONE, *A local ordered upwind method for Hamilton-Jacobi and Isaacs equations*, Proceedings of 18th IFAC World Congress 2011, pp. 6800–6805.
- [P3] S. CACACE, E. CRISTIANI, M. FALCONE AND A. PICARELLI, *A patchy dynamic programming scheme for a class of Hamilton-Jacobi-Bellman equations*, SIAM J. Sci. Comput., 34 No. 5 (2012), pp 2625–2649.
- [P4] S. CACACE, E. CRISTIANI AND M. FALCONE, *User guide for single-pass methods solving Hamilton-Jacobi equations*, 2012, in preparation.
- [P5] S. CACACE, E. CRISTIANI AND M. FALCONE, *Numerical approximation of Nash equilibria for a class of non-cooperative differential games*, L. A. Petrosjan e V. V. Mazalov (Eds.), Game Theory and Applications, vol. 16, Nova Science Publishers, New York.
- [P6] A. ALLA, M. FALCONE AND D. KALISE, *An accelerated Policy Iteration algorithm for dynamic programming*, 2012, in preparation.

References

- [1] F. ANCONA AND A. BRESSAN, *Patchy vector fields and asymptotic stabilization*, ESAIM: Control, Optimisation and Calculus of Variations, 4 (1999), pp. 445–471.
- [2] F. ANCONA AND A. BRESSAN, *Flow stability of patchy vector fields and robust feedback stabilization*, SIAM J. Control Optim., 41 (2002), pp. 1455–1476.
- [3] F. ANCONA AND A. BRESSAN, *Nearly time optimal stabilizing patchy feedbacks*, Ann. I. H. Poincaré- AN, 24 (2007), pp. 279–310.
- [4] R. J. AUMANN AND S. HART, Handbook of game theory with economic applications, vol. 2, Handbooks in Economics 11, Elsevier, North Holland, 1994.
- [5] M. BARDI AND I. CAPUZZO DOLCETTA, Optimal control and viscosity solutions of Hamilton-Jacobi-Bellman equations, Birkhäuser, Boston, 1997.

- [6] T. BASAR AND G. J. OLSDER, *Dynamic Noncooperative Game Theory*, 2nd edn. Academic, London/New York (1989). (Reprinted in SIAM Series Classics in Applied Mathematics, 1999).
- [7] R. BELLMAN, *Dynamic Programming*. Princeton University Press, Princeton, NJ, 1957.
- [8] A. BRESSAN AND F. S. PRIULI, *Infinite horizon noncooperative differential games*, J. Differential Equations, 227 (2006), pp. 230–257.
- [9] A. BRESSAN AND W. SHEN, *Small BV solutions of hyperbolic noncooperative differential games*, SIAM J. Control Optim., 43 (2004), pp. 194–215.
- [10] P. CARDALIAGUET AND S. PLASKACZ, *Existence and uniqueness of a Nash equilibrium feedback for a simple nonzero-sum differential game*, Int. J. Game Theory, 32 (2003), pp. 33–71.
- [11] E. CRISTIANI AND M. FALCONE, *Fast semi-Lagrangian schemes for the Eikonal equation and applications*, SIAM J. Numer. Anal., 45 (2007), pp. 1979–2011.
- [12] F. CAMILLI, M. FALCONE, P. LANUCARA AND A. SEGHINI, *A domain decomposition method for Bellman equations*, D.E. eyes, J. Xu, editors, Domain Decomposition methods in Scientific and Engineering Computing, Contemporary Mathematics, 180 (1994), pp. 477–483.
- [13] M. FALCONE, *Numerical solution of dynamic programming equations*, Appendix A in [5].
- [14] M. FALCONE, *Some remarks on the synthesis of feedback controls via numerical methods*, in J. L. Menaldi, E. Rofman, A. Sulem (eds.), *Optimal Control and Partial Differential Equations*, IOS Press, 2001, pp. 456–465.
- [15] F. FACCHINEI, A. FISCHER AND V. PICCIAZZI, *Generalized Nash equilibrium problems and Newton methods*, Math. Program., 117 (2009), pp. 163–194.
- [16] F. FACCHINEI AND CH. KANZOW, *Penalty methods for the solution of generalized Nash equilibrium problems*, SIAM J. Optim., 20 (2010), pp. 2228–2253.
- [17] M. FALCONE, *Numerical methods for differential games based on partial differential equations*, International Game Theory Review, 8 (2006), pp. 231–272.
- [18] M. FALCONE, P. LANUCARA AND A. SEGHINI, *A splitting algorithm for Hamilton-Jacobi-Bellman equations* *Applied Numerical Mathematics*, 15 (1994), pp. 207–218.
- [19] A. FRIEDMAN, *Differential Games*, Wiley-Interscience, New York, 1971.

- [20] M. HERMANN, *A domain decomposition parallelization of the Fast Marching Method*, in: Annual Research Briefs-2003, Center for Turbulence Research, Stanford, CA.
- [21] R.A. HOWARD, *Dynamic programming and Markov processes*. Wiley, New York, 1960.
- [22] C.Y. KAO, S. OSHER AND J. QIAN, *Lax-Friedrichs sweeping scheme for static Hamilton-Jacobi equations* J. Comput. Phys., 196 (2004), pp. 367-391.
- [23] M.L. PUTERMAN AND S.L. BRUMELLE, *On the convergence of Policy iteration in stationary Dynamic Programming*, Math. of Operation Research, 4 (1979), pp. 60-69.
- [24] A. QUARTERONI AND A. VALLI, *Domain decomposition methods for partial differential equations*, Oxford University Press, 1999.
- [25] J. A. SETHIAN, *A fast marching level set method for monotonically advancing fronts*, Proc. Natl. Acad. Sci. USA, 93 (1996), pp. 1591-1595.
- [26] J. A. SETHIAN AND A. VLADIMIRSKY, *Ordered upwind methods for static Hamilton-Jacobi equations: theory and algorithms*, SIAM J. Numer. Anal., 41 (2003), pp. 325-363.
- [27] J. N. TSITSIKLIS, *Efficient algorithms for globally optimal trajectories*, IEEE Trans. Automat. Control, 40 (1995), pp. 1528-1538.
- [28] M. C. TUGURLAN, *Fast Marching Methods – Parallel implementation and analysis*, Ph.D. Thesis, Louisiana State University, December 2008.
- [29] O. WEBER, Y. S. DEVIR, A. M. BRONSTEIN, M. M. BRONSTEIN AND R. KIMMEL, *Parallel algorithms for approximation of distance maps on parametric surfaces*, ACM Trans. Graph. 27 (2008), pp. 1-16.